

Lab 4 Requirements

Integrarea metodei cu cele mai bune rezultate din experimente într-o mini-aplicație pentru potențialul client

Tehnologii folosite

1. Backend:
 - a. Python
 - b. Flask - pentru API-uri
 - c. TensorFlow - pentru machine learning
 - d. OpenCV - pentru procesare de imagini
 - e. NumPy - pentru procesare numerică
2. Frontend:
 - a. Vue.js - framework JavaScript
 - b. HTML/CSS
 - c. JavaScript
3. Deployment:
 - a. PyInstaller - instrument pentru transformarea aplicațiilor Python în executabile
 - b. Flask-CORS - extensie pentru gestionarea cross-origin resource sharing

Etapele necesare pentru integrare

1. Pregătirea modelului
 - a. Antrenarea modelului
 - b. Salvarea modelului antrenat
2. Dezvoltarea backend-ului cu Flask
 - a. crearea unui server Flask care încarcă modelul TensorFlow antrenat, returnează predicții despre boala detectată și nivelul de confidence și servește fișiere statice pentru frontend
3. Dezvoltarea frontend-ului cu Vue.js
 - a. crearea unei interfețe intuitive care permite încărcarea imaginilor cu rodii pentru analiză și vizualizarea rezultatelor analizei
4. Integrarea frontend-backend
 - a. Construirea frontend-ului Vue.js cu **npm run build**
 - b. Copierea fișierelor construite în directorul **static** al aplicației Flask

- c. Configurarea Flask pentru a servi aplicația Vue.js ca pagină principală
5. Crearea executabilului standalone
- a. utilizarea PyInstaller pentru a împacheta aplicația într-un executabil
 - b. includerea tuturor dependențelor necesare (model, fișiere statice, biblioteci)
 - c. configurarea aplicației pentru deschiderea automată a browser-ului la pornire

Execuția aplicației:

Pentru utilizare, clientul trebuie doar să execute fișierul **HarvestGnosis.exe** generat cu PyInstaller, care va porni serverul și va deschide automat aplicația în browser.

Link cod: <https://github.com/Razvanix445/MIASC/tree/main/HarvestGnosis>

Link release: <https://github.com/Razvanix445/MIASC/releases/tag/v1.0.0>